Application of Information Theory, Lecture 11

Pseudo-Entropy and Pseudorandom Generators

Iftach Haitner

Tel Aviv University.

January 6, 2015

Part I

Motivation

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k, m)) = m$

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k,m)) = m$

- What security should we ask from such scheme?

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k,m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k,x)$.

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k, m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k, x)$.
- Theorem (Shannon): Perfect secrecy implies $\ell \geq n$.

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k, m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k, x)$.
- Theorem (Shannon): Perfect secrecy implies $\ell \geq n$.
- Is is bad?

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k, m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k, x)$.
- Theorem (Shannon): Perfect secrecy implies $\ell \geq n$.
- Is is bad?

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k, m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k, x)$.
- Theorem (Shannon): Perfect secrecy implies $\ell \geq n$.
- Is is bad? Is it optimal?

Encryption schemes

Definition 1

A pair of algorithms (E,D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k,\mathsf{E}(k,m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k,x)$.
- Theorem (Shannon): Perfect secrecy implies $\ell \geq n$.
- Is is bad? Is it optimal?
- Proof:

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k, m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k, x)$.
- Theorem (Shannon): Perfect secrecy implies $\ell \geq n$.
- Is is bad? Is it optimal?
- Proof:

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k,m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k,x)$.
- Theorem (Shannon): Perfect secrecy implies $\ell \geq n$.
- Is is bad? Is it optimal?
- Proof: Let $M \sim \{0,1\}^n$.

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k,m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k,x)$.
- Theorem (Shannon): Perfect secrecy implies $\ell \geq n$.
- Is is bad? Is it optimal?
- Proof: Let $M \sim \{0,1\}^n$.
- Perfect secrecy $\implies H(M, \mathsf{E}_K(M)) = H(M, \mathsf{E}_K(0^\ell))$

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k, m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k, x)$.
- Theorem (Shannon): Perfect secrecy implies $\ell \geq n$.
- Is is bad? Is it optimal?
- Proof: Let $M \sim \{0,1\}^n$.
- Perfect secrecy $\implies H(M, \mathsf{E}_K(M)) = H(M, \mathsf{E}_K(0^\ell))$
- $\implies H(M|\mathsf{E}_K(M)) = H(M, \mathsf{E}_K(M)) - H(\mathsf{E}_K(M)) = H(M|\mathsf{E}_K(0^\ell)) = n$

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k, m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k, x)$.
- Theorem (Shannon): Perfect secrecy implies $\ell \geq n$.
- Is is bad? Is it optimal?
- Proof: Let $M \sim \{0,1\}^n$.
- Perfect secrecy $\implies H(M, \mathsf{E}_K(M)) = H(M, \mathsf{E}_K(0^\ell))$
- $\implies H(M|\mathsf{E}_K(M)) = H(M, \mathsf{E}_K(M)) - H(\mathsf{E}_K(M)) = H(M|\mathsf{E}_K(0^\ell)) = n$
- Perfect correctness $\implies H(M|\mathsf{E}_K(M), K) = 0$

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k, m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k, x)$.
- Theorem (Shannon): Perfect secrecy implies $\ell \geq n$.
- Is is bad? Is it optimal?
- Proof: Let $M \sim \{0,1\}^n$.
- Perfect secrecy $\implies H(M, \mathsf{E}_K(M)) = H(M, \mathsf{E}_K(0^\ell))$
- $\implies H(M|\mathsf{E}_K(M)) = H(M, \mathsf{E}_K(M)) - H(\mathsf{E}_K(M)) = H(M|\mathsf{E}_K(0^\ell)) = n$
- Perfect correctness $\implies H(M|\mathsf{E}_K(M), K) = 0$
- $\implies H(M|\mathsf{E}_K(M)) \leq H(M, K|\mathsf{E}_K(M)) \leq H(K|\mathsf{E}_K(M)) + 0 \leq H(K)$

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k, m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k, x)$.
- Theorem (Shannon): Perfect secrecy implies $\ell \geq n$.
- Is is bad? Is it optimal?
- Proof: Let $M \sim \{0,1\}^n$.
- Perfect secrecy $\implies H(M, \mathsf{E}_K(M)) = H(M, \mathsf{E}_K(0^\ell))$
- $\implies H(M|\mathsf{E}_K(M)) = H(M, \mathsf{E}_K(M)) - H(\mathsf{E}_K(M)) = H(M|\mathsf{E}_K(0^\ell)) = n$
- Perfect correctness $\implies H(M|\mathsf{E}_K(M), K) = 0$
- $\implies H(M|\mathsf{E}_K(M)) \leq H(M, K|\mathsf{E}_K(M)) \leq H(K|\mathsf{E}_K(M)) + 0 \leq H(K)$
- $\implies n \leq \ell. \square$

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k, m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k, x)$.
- Theorem (Shannon): Perfect secrecy implies $\ell \geq n$.
- Is is bad? Is it optimal?
- Proof: Let $M \sim \{0,1\}^n$.
- Perfect secrecy $\implies H(M, \mathsf{E}_K(M)) = H(M, \mathsf{E}_K(0^\ell))$
- $\implies H(M|\mathsf{E}_K(M)) = H(M, \mathsf{E}_K(M)) - H(\mathsf{E}_K(M)) = H(M|\mathsf{E}_K(0^\ell)) = n$
- Perfect correctness $\implies H(M|\mathsf{E}_K(M), K) = 0$
- $\implies H(M|\mathsf{E}_K(M)) \leq H(M, K|\mathsf{E}_K(M)) \leq H(K|\mathsf{E}_K(M)) + 0 \leq H(K)$
- $\implies n \leq \ell. \square$
- Statistical security?

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k, m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k, x)$.
- Theorem (Shannon): Perfect secrecy implies $\ell \geq n$.
- Is is bad? Is it optimal?
- Proof: Let $M \sim \{0,1\}^n$.
- Perfect secrecy $\implies H(M, \mathsf{E}_K(M)) = H(M, \mathsf{E}_K(0^\ell))$
- $\implies H(M|\mathsf{E}_K(M)) = H(M, \mathsf{E}_K(M)) - H(\mathsf{E}_K(M)) = H(M|\mathsf{E}_K(0^\ell)) = n$
- Perfect correctness $\implies H(M|\mathsf{E}_K(M), K) = 0$
- $\implies H(M|\mathsf{E}_K(M)) \leq H(M, K|\mathsf{E}_K(M)) \leq H(K|\mathsf{E}_K(M)) + 0 \leq H(K)$
- $\implies n \leq \ell. \square$
- Statistical security?

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k, m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k, x)$.
- Theorem (Shannon): Perfect secrecy implies $\ell \geq n$.
- Is is bad? Is it optimal?
- Proof: Let $M \sim \{0,1\}^n$.
- Perfect secrecy $\implies H(M, \mathsf{E}_K(M)) = H(M, \mathsf{E}_K(0^\ell))$
- $\implies H(M|\mathsf{E}_K(M)) = H(M, \mathsf{E}_K(M)) - H(\mathsf{E}_K(M)) = H(M|\mathsf{E}_K(0^\ell)) = n$
- Perfect correctness $\implies H(M|\mathsf{E}_K(M), K) = 0$
- $\implies H(M|\mathsf{E}_K(M)) \leq H(M, K|\mathsf{E}_K(M)) \leq H(K|\mathsf{E}_K(M)) + 0 \leq H(K)$
- $\implies n \leq \ell. \square$
- Statistical security? HW.

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0,1\}^n$ and $m \in \{0,1\}^\ell$, it holds that $\mathsf{D}(k, \mathsf{E}(k, m)) = m$

- What security should we ask from such scheme?
- Perfect secrecy: $\mathsf{E}_K(m) \equiv \mathsf{E}_K(m')$, for any $m, m' \in \{0,1\}^\ell$ and $K \sim \{0,1\}^n$, letting $\mathsf{E}_k(x) := \mathsf{E}(k, x)$.
- Theorem (Shannon): Perfect secrecy implies $\ell \geq n$.
- Is is bad? Is it optimal?
- Proof: Let $M \sim \{0,1\}^n$.
- Perfect secrecy $\implies H(M, \mathsf{E}_K(M)) = H(M, \mathsf{E}_K(0^\ell))$
- $\implies H(M|\mathsf{E}_K(M)) = H(M, \mathsf{E}_K(M)) - H(\mathsf{E}_K(M)) = H(M|\mathsf{E}_K(0^\ell)) = n$
- Perfect correctness $\implies H(M|\mathsf{E}_K(M), K) = 0$
- $\implies H(M|\mathsf{E}_K(M)) \leq H(M, K|\mathsf{E}_K(M)) \leq H(K|\mathsf{E}_K(M)) + 0 \leq H(K)$
- $\implies n \leq \ell. \square$
- Statistical security? HW. Computational security?

Part II

Statistical Vs. Computational distance

Distributions and statistical distance

Let $\mathcal{P}$ and $\mathcal{Q}$ be two distributions over a finite set $\mathcal{U}$. Their statistical distance (also known as, variation distance) is defined as

$$\mathsf{SD}(\mathcal{P},\mathcal{Q}) := \frac{1}{2}\sum_{x\in\mathcal{U}} |\mathcal{P}(x) - \mathcal{Q}(x)| = \max_{\mathcal{S}\subseteq\mathcal{U}} (\mathcal{P}(\mathcal{S}) - \mathcal{Q}(\mathcal{S}))$$

We will only consider finite distributions.

Distributions and statistical distance

Let $\mathcal{P}$ and $\mathcal{Q}$ be two distributions over a finite set $\mathcal{U}$. Their statistical distance (also known as, variation distance) is defined as

$$\mathsf{SD}(\mathcal{P}, \mathcal{Q}) := \frac{1}{2} \sum_{x \in \mathcal{U}} |\mathcal{P}(x) - \mathcal{Q}(x)| = \max_{\mathcal{S} \subseteq \mathcal{U}} (\mathcal{P}(\mathcal{S}) - \mathcal{Q}(\mathcal{S}))$$

We will only consider finite distributions.

Claim 2

For any pair of (finite) distributions $\mathcal{P}$ and $\mathcal{Q}$, it holds that

$$\mathsf{SD}(\mathcal{P}, \mathcal{Q}) = \max_{\mathsf{D}} \{\Delta^{\mathsf{D}}(\mathcal{P}, \mathcal{Q}) := \Pr_{x \leftarrow \mathcal{P}}[\mathsf{D}(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}}[\mathsf{D}(x) = 1]\},$$

where D is any algorithm.

Distributions and statistical distance

Let $\mathcal{P}$ and $\mathcal{Q}$ be two distributions over a finite set $\mathcal{U}$. Their statistical distance (also known as, variation distance) is defined as

$$\mathsf{SD}(\mathcal{P}, \mathcal{Q}) := \frac{1}{2} \sum_{x \in \mathcal{U}} |\mathcal{P}(x) - \mathcal{Q}(x)| = \max_{\mathcal{S} \subseteq \mathcal{U}} (\mathcal{P}(\mathcal{S}) - \mathcal{Q}(\mathcal{S}))$$

We will only consider finite distributions.

Claim 2

For any pair of (finite) distributions $\mathcal{P}$ and $\mathcal{Q}$, it holds that

$$\mathsf{SD}(\mathcal{P}, \mathcal{Q}) = \max_{\mathsf{D}} \{\Delta^{\mathsf{D}}(\mathcal{P}, \mathcal{Q}) := \Pr_{x \leftarrow \mathcal{P}} [\mathsf{D}(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}} [\mathsf{D}(x) = 1]\},$$

where D is any algorithm.

Let $\mathcal{P}, \mathcal{Q}, R$ be finite distributions, then

Triangle inequality: $\mathsf{SD}(\mathcal{P}, R) \leq \mathsf{SD}(\mathcal{P}, \mathcal{Q}) + \mathsf{SD}(\mathcal{Q}, R)$

Repeated sampling: $\mathsf{SD}(\mathcal{P}^2 = (\mathcal{P}, \mathcal{P}), \mathcal{Q}^2 = (\mathcal{Q}, \mathcal{Q})) \leq 2 \cdot \mathsf{SD}(\mathcal{P}, \mathcal{Q})$

Section 1

Computational Indistinguishability

Computational indistinguishability

Definition 3 (computational indistinguishability)

$\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, if $\Delta^{\mathsf{D}}_{\mathcal{P},\mathcal{Q}} \leq \varepsilon$, for any s-size D.

Computational indistinguishability

Definition 3 (computational indistinguishability)

$\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, if $\Delta^{\mathsf{D}}_{\mathcal{P},\mathcal{Q}} \leq \varepsilon$, for any s-size D.

- Adversaries are circuits (possibly randomized)

Computational indistinguishability

Definition 3 (computational indistinguishability)

$\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, if $\Delta^{\mathsf{D}}_{\mathcal{P},\mathcal{Q}} \leq \varepsilon$, for any s-size D.

- Adversaries are circuits (possibly randomized)
- (∞, ε)-indistinguishable is equivalent to statistical distance ε

Computational indistinguishability

Definition 3 (computational indistinguishability)

$\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, if $\Delta^{\mathsf{D}}_{\mathcal{P},\mathcal{Q}} \leq \varepsilon$, for any s-size D.

- Adversaries are circuits (possibly randomized)
- (∞, ε)-indistinguishable is equivalent to statistical distance ε
- We sometimes think of $s = n^{\omega(1)}$ and $\varepsilon = 1/s$, where n is the “security parameter”

Computational indistinguishability

Definition 3 (computational indistinguishability)

$\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, if $\Delta^{\mathsf{D}}_{\mathcal{P},\mathcal{Q}} \leq \varepsilon$, for any s-size D.

- Adversaries are circuits (possibly randomized)
- (∞, ε)-indistinguishable is equivalent to statistical distance ε
- We sometimes think of $s = n^{\omega(1)}$ and $\varepsilon = 1/s$, where n is the "security parameter"
- Can it be different from the statistical case?

Computational indistinguishability

Definition 3 (computational indistinguishability)

$\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, if $\Delta^{\mathsf{D}}_{\mathcal{P},\mathcal{Q}} \leq \varepsilon$, for any s-size D.

- Adversaries are circuits (possibly randomized)
- (∞, ε)-indistinguishable is equivalent to statistical distance ε
- We sometimes think of $s = n^{\omega(1)}$ and $\varepsilon = 1/s$, where n is the "security parameter"
- Can it be different from the statistical case?
- Unless said otherwise, distributions are over $\{0, 1\}^n$

Repeated sampling

Question 4

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, what about $\mathcal{P}^2$ and $\mathcal{Q}^2$?

Repeated sampling

Question 4

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, what about $\mathcal{P}^2$ and $\mathcal{Q}^2$?

- Let D be an s'-size algorithm with $\Delta^{\mathsf{D}}(\mathcal{P}^2, \mathcal{Q}^2) = \varepsilon'$

Repeated sampling

Question 4

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, what about $\mathcal{P}^2$ and $\mathcal{Q}^2$?

- Let D be an s'-size algorithm with $\Delta^{\mathsf{D}}(\mathcal{P}^2, \mathcal{Q}^2) = \varepsilon'$

Repeated sampling

Question 4

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, what about $\mathcal{P}^2$ and $\mathcal{Q}^2$?

- Let D be an s'-size algorithm with $\Delta^{\mathrm{D}}(\mathcal{P}^2, \mathcal{Q}^2) = \varepsilon'$

$$\begin{aligned}
\varepsilon' &= \Pr_{x \leftarrow \mathcal{P}^2}[\mathrm{D}(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}^2}[\mathrm{D}(x) = 1] \\
&= \big(\Pr_{x \leftarrow \mathcal{P}^2}[\mathrm{D}(x) = 1] - \Pr_{x \leftarrow (\mathcal{P},\mathcal{Q})}[\mathrm{D}(x) = 1]\big) \\
&\quad + \big(\Pr_{x \leftarrow (\mathcal{P},\mathcal{Q})}[\mathrm{D}(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}^2}[\mathrm{D}(x) = 1]\big)
\end{aligned}$$

Repeated sampling

Question 4

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, what about $\mathcal{P}^2$ and $\mathcal{Q}^2$?

- Let D be an s'-size algorithm with $\Delta^{\mathsf{D}}(\mathcal{P}^2, \mathcal{Q}^2) = \varepsilon'$

$$\begin{aligned}
\varepsilon' &= \Pr_{x \leftarrow \mathcal{P}^2}[\mathsf{D}(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}^2}[\mathsf{D}(x) = 1] \\
&= \Big(\Pr_{x \leftarrow \mathcal{P}^2}[\mathsf{D}(x) = 1] - \Pr_{x \leftarrow (\mathcal{P},\mathcal{Q})}[\mathsf{D}(x) = 1]\Big) \\
&\quad + \Big(\Pr_{x \leftarrow (\mathcal{P},\mathcal{Q})}[\mathsf{D}(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}^2}[\mathsf{D}(x) = 1]\Big) \\
&= \Delta^{\mathsf{D}}_{(\mathcal{P}^2,(\mathcal{P},\mathcal{Q})} + \Delta^{\mathsf{D}}_{((\mathcal{P},\mathcal{Q}),\mathcal{Q}^2)}
\end{aligned}$$

Repeated sampling

Question 4

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, what about $\mathcal{P}^2$ and $\mathcal{Q}^2$?

- Let D be an s'-size algorithm with $\Delta^{\mathsf{D}}(\mathcal{P}^2, \mathcal{Q}^2) = \varepsilon'$

$$
\begin{aligned}
\varepsilon' &= \Pr_{x \leftarrow \mathcal{P}^2}[\mathsf{D}(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}^2}[\mathsf{D}(x) = 1] \\
&= \Big(\Pr_{x \leftarrow \mathcal{P}^2}[\mathsf{D}(x) = 1] - \Pr_{x \leftarrow (\mathcal{P},\mathcal{Q})}[\mathsf{D}(x) = 1]\Big) \\
&\quad + \Big(\Pr_{x \leftarrow (\mathcal{P},\mathcal{Q})}[\mathsf{D}(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}^2}[\mathsf{D}(x) = 1]\Big) \\
&= \Delta^{\mathsf{D}}_{(\mathcal{P}^2,(\mathcal{P},\mathcal{Q})} + \Delta^{\mathsf{D}}_{((\mathcal{P},\mathcal{Q}),\mathcal{Q}^2)}
\end{aligned}
$$

- So either $\Delta^{\mathsf{D}}_{(\mathcal{P}^2,(\mathcal{P},\mathcal{Q})} \geq \varepsilon'/2$, or $\Delta^{\mathsf{D}}_{((\mathcal{P},\mathcal{Q}),\mathcal{Q}^2)} \geq \varepsilon'/2$

Repeated sampling

Question 4

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, what about $\mathcal{P}^2$ and $\mathcal{Q}^2$?

- Let D be an s'-size algorithm with $\Delta^{\mathrm{D}}(\mathcal{P}^2, \mathcal{Q}^2) = \varepsilon'$

$$
\begin{aligned}
\varepsilon' &= \Pr_{x \leftarrow \mathcal{P}^2}[\mathrm{D}(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}^2}[\mathrm{D}(x) = 1] \\
&= (\Pr_{x \leftarrow \mathcal{P}^2}[\mathrm{D}(x) = 1] - \Pr_{x \leftarrow (\mathcal{P}, \mathcal{Q})}[\mathrm{D}(x) = 1]) \\
&\quad + (\Pr_{x \leftarrow (\mathcal{P}, \mathcal{Q})}[\mathrm{D}(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}^2}[\mathrm{D}(x) = 1]) \\
&= \Delta^{\mathrm{D}}_{(\mathcal{P}^2, (\mathcal{P}, \mathcal{Q})} + \Delta^{\mathrm{D}}_{((\mathcal{P}, \mathcal{Q}), \mathcal{Q}^2)}
\end{aligned}
$$

- So either $\Delta^{\mathrm{D}}_{(\mathcal{P}^2, (\mathcal{P}, \mathcal{Q})} \geq \varepsilon'/2$, or $\Delta^{\mathrm{D}}_{((\mathcal{P}, \mathcal{Q}), \mathcal{Q}^2)} \geq \varepsilon'/2$
- Hence, $\varepsilon' < 2\varepsilon$ implies $s' \geq s - 2n$.

Repeated sampling cont.

What about $\mathcal{P}^k$ and $\mathcal{Q}^k$?

Repeated sampling cont.

What about $\mathcal{P}^k$ and $\mathcal{Q}^k$?

Claim 5

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, then $\mathcal{P}^k$ and $\mathcal{Q}^k$ are $(s - kn, k\varepsilon)$-indistinguishable.

Repeated sampling cont.

What about $\mathcal{P}^k$ and $\mathcal{Q}^k$?

Claim 5

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, then $\mathcal{P}^k$ and $\mathcal{Q}^k$ are $(s - kn, k\varepsilon)$-indistinguishable.

Proof: ?

Repeated sampling cont.

What about $\mathcal{P}^k$ and $\mathcal{Q}^k$?

Claim 5

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, then $\mathcal{P}^k$ and $\mathcal{Q}^k$ are $(s - kn, k\varepsilon)$-indistinguishable.

Proof: ?

- For $i \in \{0, \ldots, k\}$, let $H^i = (P_1, \ldots, P_i, Q_{i+1}, \ldots, Q_k)$, where the P_i's are iid $\sim \mathcal{P}$ and the Q_i's are iid $\sim \mathcal{Q}$. (hybrids)

Repeated sampling cont.

What about $\mathcal{P}^k$ and $\mathcal{Q}^k$?

Claim 5

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, then $\mathcal{P}^k$ and $\mathcal{Q}^k$ are $(s - kn, k\varepsilon)$-indistinguishable.

Proof: ?

- For $i \in \{0, \ldots, k\}$, let $H^i = (P_1, \ldots, P_i, Q_{i+1}, \ldots, Q_k)$, where the P_i's are iid $\sim \mathcal{P}$ and the Q_i's are iid $\sim \mathcal{Q}$. (hybrids)
- Let D be a s'-size algorithm with $\Delta^{\mathsf{D}}(\mathcal{P}^k, \mathcal{Q}^k) = \varepsilon'$

Repeated sampling cont.

What about $\mathcal{P}^k$ and $\mathcal{Q}^k$?

Claim 5

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, then $\mathcal{P}^k$ and $\mathcal{Q}^k$ are $(s - kn, k\varepsilon)$-indistinguishable.

Proof: ?

- For $i \in \{0, \ldots, k\}$, let $H^i = (P_1, \ldots, P_i, Q_{i+1}, \ldots, Q_k)$, where the P_i's are iid $\sim \mathcal{P}$ and the Q_i's are iid $\sim \mathcal{Q}$. (hybrids)
- Let D be a s'-size algorithm with $\Delta^{\mathsf{D}}(\mathcal{P}^k, \mathcal{Q}^k) = \varepsilon'$
- $\varepsilon' = \Pr\left[\mathsf{D}(H^k) = 1\right] - \Pr\left[\mathsf{D}(H^0) = 1\right]$.

Repeated sampling cont.

What about $\mathcal{P}^k$ and $\mathcal{Q}^k$?

Claim 5

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, then $\mathcal{P}^k$ and $\mathcal{Q}^k$ are $(s - kn, k\varepsilon)$-indistinguishable.

Proof: ?

- For $i \in \{0, \ldots, k\}$, let $H^i = (P_1, \ldots, P_i, Q_{i+1}, \ldots, Q_k)$, where the P_i's are iid $\sim \mathcal{P}$ and the Q_i's are iid $\sim \mathcal{Q}$. (hybrids)
- Let D be a s'-size algorithm with $\Delta^{\mathsf{D}}(\mathcal{P}^k, \mathcal{Q}^k) = \varepsilon'$
- $\varepsilon' = \Pr\left[\mathsf{D}(H^k) = 1\right] - \Pr\left[\mathsf{D}(H^0) = 1\right]$.
- $\varepsilon' = \sum_{i \in [k]} \Pr\left[\mathsf{D}(H^i) = 1\right] - \Pr\left[\mathsf{D}(H^{i-1}) = 1\right]$

Repeated sampling cont.

What about $\mathcal{P}^k$ and $\mathcal{Q}^k$?

Claim 5

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, then $\mathcal{P}^k$ and $\mathcal{Q}^k$ are $(s - kn, k\varepsilon)$-indistinguishable.

Proof: ?

- For $i \in \{0, \ldots, k\}$, let $H^i = (P_1, \ldots, P_i, Q_{i+1}, \ldots, Q_k)$, where the P_i's are iid $\sim \mathcal{P}$ and the Q_i's are iid $\sim \mathcal{Q}$. (hybrids)
- Let D be a s'-size algorithm with $\Delta^{\mathsf{D}}(\mathcal{P}^k, \mathcal{Q}^k) = \varepsilon'$
- $\varepsilon' = \Pr\left[\mathsf{D}(H^k) = 1\right] - \Pr\left[\mathsf{D}(H^0) = 1\right]$.
- $\varepsilon' = \sum_{i \in [k]} \Pr\left[\mathsf{D}(H^i) = 1\right] - \Pr\left[\mathsf{D}(H^{i-1}) = 1\right]$

Repeated sampling cont.

What about $\mathcal{P}^k$ and $\mathcal{Q}^k$?

Claim 5

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, then $\mathcal{P}^k$ and $\mathcal{Q}^k$ are $(s - kn, k\varepsilon)$-indistinguishable.

Proof: ?

- For $i \in \{0, \ldots, k\}$, let $H^i = (P_1, \ldots, P_i, Q_{i+1}, \ldots, Q_k)$, where the P_i's are iid $\sim \mathcal{P}$ and the Q_i's are iid $\sim \mathcal{Q}$. (hybrids)
- Let D be a s'-size algorithm with $\Delta^{\mathsf{D}}(\mathcal{P}^k, \mathcal{Q}^k) = \varepsilon'$
- $\varepsilon' = \Pr\left[\mathsf{D}(H^k) = 1\right] - \Pr\left[\mathsf{D}(H^0) = 1\right]$.
- $\varepsilon' = \sum_{i\in[k]} \Pr\left[\mathsf{D}(H^i) = 1\right] - \Pr\left[\mathsf{D}(H^{i-1}) = 1\right] = \sum_{i\in[k]} \Delta^{\mathsf{D}}(H^i, H^{i-1})$
- $\implies \exists i \in [k]$ with $\Delta^{\mathsf{D}}(H^i, H^{i-1}) \geq \varepsilon'/k$.

Repeated sampling cont.

What about $\mathcal{P}^k$ and $\mathcal{Q}^k$?

Claim 5

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, then $\mathcal{P}^k$ and $\mathcal{Q}^k$ are $(s - kn, k\varepsilon)$-indistinguishable.

Proof: ?

- For $i \in \{0, \ldots, k\}$, let $H^i = (P_1, \ldots, P_i, Q_{i+1}, \ldots, Q_k)$, where the P_i's are iid $\sim \mathcal{P}$ and the Q_i's are iid $\sim \mathcal{Q}$. (hybrids)
- Let D be a s'-size algorithm with $\Delta^{\mathrm{D}}(\mathcal{P}^k, \mathcal{Q}^k) = \varepsilon'$
- $\varepsilon' = \Pr\left[\mathrm{D}(H^k) = 1\right] - \Pr\left[\mathrm{D}(H^0) = 1\right]$.
- $\varepsilon' = \sum_{i \in [k]} \Pr\left[\mathrm{D}(H^i) = 1\right] - \Pr\left[\mathrm{D}(H^{i-1}) = 1\right] = \sum_{i \in [k]} \Delta^{\mathrm{D}}(H^i, H^{i-1})$
- $\implies \exists i \in [k]$ with $\Delta^{\mathrm{D}}(H^i, H^{i-1}) \geq \varepsilon'/k$.
- Thus, $\varepsilon' \leq k\varepsilon$ implies $s' > s - kn$

Repeated sampling cont.

What about $\mathcal{P}^k$ and $\mathcal{Q}^k$?

Claim 5

Assume $\mathcal{P}$ and $\mathcal{Q}$ are (s, ε)-indistinguishable, then $\mathcal{P}^k$ and $\mathcal{Q}^k$ are $(s - kn, k\varepsilon)$-indistinguishable.

Proof: ?

- For $i \in \{0, \ldots, k\}$, let $H^i = (P_1, \ldots, P_i, Q_{i+1}, \ldots, Q_k)$, where the P_i's are iid $\sim \mathcal{P}$ and the Q_i's are iid $\sim \mathcal{Q}$. (hybrids)
- Let D be a s'-size algorithm with $\Delta^{\mathsf{D}}(\mathcal{P}^k, \mathcal{Q}^k) = \varepsilon'$
- $\varepsilon' = \Pr\left[\mathsf{D}(H^k) = 1\right] - \Pr\left[\mathsf{D}(H^0) = 1\right]$.
- $\varepsilon' = \sum_{i \in [k]} \Pr\left[\mathsf{D}(H^i) = 1\right] - \Pr\left[\mathsf{D}(H^{i-1}) = 1\right] = \sum_{i \in [k]} \Delta^{\mathsf{D}}(H^i, H^{i-1})$
- $\implies \exists i \in [k]$ with $\Delta^{\mathsf{D}}(H^i, H^{i-1}) \geq \varepsilon'/k$.
- Thus, $\varepsilon' \leq k\varepsilon$ implies $s' > s - kn$
- When considering bounded time algorithms, things behaves very differently!

Part III

Pseudorandom Generators

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution $\mathcal{P}$ over $\{0,1\}^n$ is (s,ε)-pseudorandom, if it is (s,ε)-indistinguishable from U_n.

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution $\mathcal{P}$ over $\{0,1\}^n$ is (s,ε)-pseudorandom, if it is (s,ε)-indistinguishable from U_n.

- Do such distributions exit for interesting (s,ε)

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution $\mathcal{P}$ over $\{0,1\}^n$ is (s,ε)-pseudorandom, if it is (s,ε)-indistinguishable from U_n.

- Do such distributions exit for interesting (s,ε)

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution $\mathcal{P}$ over $\{0,1\}^n$ is (s,ε)-pseudorandom, if it is (s,ε)-indistinguishable from U_n.

- Do such distributions exit for interesting (s,ε)

Definition 7 (pseudorandom generators (PRGs))

A poly-time computable function $g\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ is a (s,ε)-pseudorandom generator, if for any $n \in \mathbb{N}$

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution $\mathcal{P}$ over $\{0,1\}^n$ is (s,ε)-pseudorandom, if it is (s,ε)-indistinguishable from U_n.

- Do such distributions exit for interesting (s,ε)

Definition 7 (pseudorandom generators (PRGs))

A poly-time computable function $g\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ is a (s,ε)-pseudorandom generator, if for any $n \in \mathbb{N}$

- g is length extending (i.e., $\ell(n) > n$)

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution $\mathcal{P}$ over $\{0,1\}^n$ is (s,ε)-pseudorandom, if it is (s,ε)-indistinguishable from U_n.

- Do such distributions exit for interesting (s,ε)

Definition 7 (pseudorandom generators (PRGs))

A poly-time computable function $g\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ is a (s,ε)-pseudorandom generator, if for any $n \in \mathbb{N}$

- g is length extending (i.e., $\ell(n) > n$)
- $g(U_n)$ is $(s(n),\varepsilon(n))$-pseudorandom

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution $\mathcal{P}$ over $\{0,1\}^n$ is (s,ε)-pseudorandom, if it is (s,ε)-indistinguishable from U_n.

- Do such distributions exit for interesting (s,ε)

Definition 7 (pseudorandom generators (PRGs))

A poly-time computable function $g\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ is a (s,ε)-pseudorandom generator, if for any $n \in \mathbb{N}$

- g is length extending (i.e., $\ell(n) > n$)
- $g(U_n)$ is $(s(n),\varepsilon(n))$-pseudorandom

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution $\mathcal{P}$ over $\{0,1\}^n$ is (s,ε)-pseudorandom, if it is (s,ε)-indistinguishable from U_n.

- Do such distributions exit for interesting (s,ε)

Definition 7 (pseudorandom generators (PRGs))

A poly-time computable function $g\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ is a (s,ε)-pseudorandom generator, if for any $n \in \mathbb{N}$

- g is length extending (i.e., $\ell(n) > n$)
- $g(U_n)$ is $(s(n),\varepsilon(n))$-pseudorandom

- We omit the “security parameter", i.e., n, when its value is clear from the context

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution $\mathcal{P}$ over $\{0,1\}^n$ is (s,ε)-pseudorandom, if it is (s,ε)-indistinguishable from U_n.

- Do such distributions exit for interesting (s,ε)

Definition 7 (pseudorandom generators (PRGs))

A poly-time computable function $g\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ is a (s,ε)-pseudorandom generator, if for any $n \in \mathbb{N}$

- g is length extending (i.e., $\ell(n) > n$)
- $g(U_n)$ is $(s(n),\varepsilon(n))$-pseudorandom

- We omit the “security parameter", i.e., n, when its value is clear from the context
- Do such generators exist?

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution $\mathcal{P}$ over $\{0,1\}^n$ is (s,ε)-pseudorandom, if it is (s,ε)-indistinguishable from U_n.

- Do such distributions exit for interesting (s,ε)

Definition 7 (pseudorandom generators (PRGs))

A poly-time computable function $g\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ is a (s,ε)-pseudorandom generator, if for any $n \in \mathbb{N}$

- g is length extending (i.e., $\ell(n) > n$)
- $g(U_n)$ is $(s(n),\varepsilon(n))$-pseudorandom

- We omit the “security parameter", i.e., n, when its value is clear from the context
- Do such generators exist?
- Applications?

Section 2

Pseudorandom generators (PRGs) from One-Way Permutations (OWPs)

OWP to PRG

Claim 8

Let $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ be a poly-time permutation and let $b\colon \{0,1\}^n \mapsto \{0,1\}$ be a poly-time (s,ε)-hardcore predicate of f, then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$-PRG.

OWP to PRG

Claim 8

Let $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ be a poly-time permutation and let $b\colon \{0,1\}^n \mapsto \{0,1\}$ be a poly-time (s,ε)-hardcore predicate of f, then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$-PRG.

- Hence, OWP $\implies$ PRG

OWP to PRG

Claim 8

Let $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ be a poly-time permutation and let $b\colon \{0,1\}^n \mapsto \{0,1\}$ be a poly-time (s,ε)-hardcore predicate of f, then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$-PRG.

- Hence, OWP $\implies$ PRG
- Proof: Let D be an s'-size algorithm with $\Delta^{\mathsf{D}}_{g(U_n),U_{n+1}} = \varepsilon'$, we will show $\exists$ $(s' + O(n))$-size P with $\Pr\left[\mathsf{P}(f(U_n)) = b(U_n)\right] = \frac{1}{2} + \varepsilon'$.

OWP to PRG

Claim 8

Let $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ be a poly-time permutation and let $b\colon \{0,1\}^n \mapsto \{0,1\}$ be a poly-time (s,ε)-hardcore predicate of f, then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$-PRG.

- Hence, OWP $\implies$ PRG
- Proof: Let D be an s'-size algorithm with $\Delta^{\mathsf{D}}_{g(U_n),U_{n+1}} = \varepsilon'$, we will show $\exists$ $(s' + O(n))$-size P with $\Pr[\mathsf{P}(f(U_n)) = b(U_n)] = \frac{1}{2} + \varepsilon'$.
- Let $\delta = \Pr[\mathsf{D}(U_{n+1}) = 1]$ (hence, $\Pr[\mathsf{D}(g(U_n)) = 1] = \delta + \varepsilon'$)

OWP to PRG

Claim 8

Let $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ be a poly-time permutation and let $b\colon \{0,1\}^n \mapsto \{0,1\}$ be a poly-time (s,ε)-hardcore predicate of f, then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$-PRG.

- Hence, OWP $\implies$ PRG
- Proof: Let D be an s'-size algorithm with $\Delta^{\mathsf{D}}_{g(U_n),U_{n+1}} = \varepsilon'$, we will show $\exists$ $(s' + O(n))$-size P with $\Pr[\mathsf{P}(f(U_n)) = b(U_n)] = \frac{1}{2} + \varepsilon'$.
- Let $\delta = \Pr[\mathsf{D}(U_{n+1}) = 1]$ (hence, $\Pr[\mathsf{D}(g(U_n)) = 1] = \delta + \varepsilon'$)
- Compute

$$\delta = \Pr[\mathsf{D}(f(U_n), U_1) = 1]$$

OWP to PRG

Claim 8

Let $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ be a poly-time permutation and let $b\colon \{0,1\}^n \mapsto \{0,1\}$ be a poly-time (s,ε)-hardcore predicate of f, then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$-PRG.

- Hence, OWP $\implies$ PRG
- Proof: Let D be an s'-size algorithm with $\Delta^{\mathsf{D}}_{g(U_n),U_{n+1}} = \varepsilon'$, we will show $\exists$ $(s' + O(n))$-size P with $\Pr\left[\mathsf{P}(f(U_n)) = b(U_n)\right] = \frac{1}{2} + \varepsilon'$.
- Let $\delta = \Pr\left[\mathsf{D}(U_{n+1}) = 1\right]$ (hence, $\Pr\left[\mathsf{D}(g(U_n)) = 1\right] = \delta + \varepsilon'$)
- Compute

$$\delta = \Pr[\mathsf{D}(f(U_n), U_1) = 1]$$

OWP to PRG

Claim 8

Let $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ be a poly-time permutation and let $b\colon \{0,1\}^n \mapsto \{0,1\}$ be a poly-time (s,ε)-hardcore predicate of f, then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$-PRG.

- Hence, OWP $\implies$ PRG
- Proof: Let D be an s'-size algorithm with $\Delta^{\mathsf{D}}_{g(U_n),U_{n+1}} = \varepsilon'$, we will show $\exists$ $(s' + O(n))$-size P with $\Pr[\mathsf{P}(f(U_n)) = b(U_n)] = \frac{1}{2} + \varepsilon'$.
- Let $\delta = \Pr[\mathsf{D}(U_{n+1}) = 1]$ (hence, $\Pr[\mathsf{D}(g(U_n)) = 1] = \delta + \varepsilon'$)
- Compute

$$\begin{aligned}\delta =& \Pr[\mathsf{D}(f(U_n), U_1) = 1] \\ &= \Pr[U_1 = b(U_n)] \cdot \Pr[\mathsf{D}(f(U_n), U_1) = 1 \mid U_1 = b(U_n)] \\ &+ \Pr[U_1 = \overline{b(U_n)}] \cdot \Pr[\mathsf{D}(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}]\end{aligned}$$

OWP to PRG

Claim 8

Let $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ be a poly-time permutation and let $b\colon \{0,1\}^n \mapsto \{0,1\}$ be a poly-time (s,ε)-hardcore predicate of f, then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$-PRG.

- Hence, OWP $\implies$ PRG
- Proof: Let D be an s'-size algorithm with $\Delta^{\mathsf{D}}_{g(U_n),U_{n+1}} = \varepsilon'$, we will show $\exists$ $(s' + O(n))$-size P with $\Pr\left[\mathsf{P}(f(U_n)) = b(U_n)\right] = \frac{1}{2} + \varepsilon'$.
- Let $\delta = \Pr\left[\mathsf{D}(U_{n+1}) = 1\right]$ (hence, $\Pr\left[\mathsf{D}(g(U_n)) = 1\right] = \delta + \varepsilon'$)
- Compute

$$\begin{aligned}
\delta &= \Pr[\mathsf{D}(f(U_n), U_1) = 1] \\
&= \Pr[U_1 = b(U_n)] \cdot \Pr[\mathsf{D}(f(U_n), U_1) = 1 \mid U_1 = b(U_n)] \\
&\quad + \Pr[U_1 = \overline{b(U_n)}] \cdot \Pr[\mathsf{D}(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}] \\
&= \frac{1}{2}(\delta + \varepsilon') + \frac{1}{2} \cdot \Pr[\mathsf{D}(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}].
\end{aligned}$$

OWP to PRG

Claim 8

Let $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ be a poly-time permutation and let $b\colon \{0,1\}^n \mapsto \{0,1\}$ be a poly-time (s,ε)-hardcore predicate of f, then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$-PRG.

- Hence, OWP $\implies$ PRG
- Proof: Let D be an s'-size algorithm with $\Delta^{\mathsf{D}}_{g(U_n),U_{n+1}} = \varepsilon'$, we will show $\exists$ $(s' + O(n))$-size P with $\Pr\left[\mathsf{P}(f(U_n)) = b(U_n)\right] = \frac{1}{2} + \varepsilon'$.
- Let $\delta = \Pr\left[\mathsf{D}(U_{n+1}) = 1\right]$ (hence, $\Pr\left[\mathsf{D}(g(U_n)) = 1\right] = \delta + \varepsilon'$)
- Compute

$$\begin{aligned}
\delta = & \Pr[\mathsf{D}(f(U_n), U_1) = 1] \\
& = \Pr[U_1 = b(U_n)] \cdot \Pr[\mathsf{D}(f(U_n), U_1) = 1 \mid U_1 = b(U_n)] \\
& + \Pr[U_1 = \overline{b(U_n)}] \cdot \Pr[\mathsf{D}(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}] \\
& = \frac{1}{2}(\delta + \varepsilon') + \frac{1}{2} \cdot \Pr[\mathsf{D}(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}].
\end{aligned}$$

- Hence, $\Pr\left[\mathsf{D}(f(U_n), \overline{b(U_n)}) = 1\right] = \delta - \varepsilon'$

OWP to PRG cont.

- $\Pr[\mathsf{D}(f(U_n), b(U_n)) = 1] = \delta + \varepsilon'$
- $\Pr[\mathsf{D}(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon'$

OWP to PRG cont.

- $\Pr[\mathrm{D}(f(U_n), b(U_n)) = 1] = \delta + \varepsilon'$
- $\Pr[\mathrm{D}(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon'$

Algorithm 9 (P)

Input: $y \in \{0,1\}^n$

1. Flip a random coin $c \leftarrow \{0,1\}$.
2. If $\mathrm{D}(y, c) = 1$ output c, otherwise, output $\overline{c}$.

OWP to PRG cont.

- $\Pr[\mathrm{D}(f(U_n), b(U_n)) = 1] = \delta + \varepsilon'$
- $\Pr[\mathrm{D}(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon'$

Algorithm 9 (P)

Input: $y \in \{0,1\}^n$

1. Flip a random coin $c \leftarrow \{0,1\}$.
2. If $\mathrm{D}(y, c) = 1$ output c, otherwise, output $\overline{c}$.

- It follows that

$$\begin{aligned}\Pr\left[\mathrm{P}(f(U_n)) = b(U_n)\right] = \Pr[c = b(U_n)] \cdot \Pr\left[\mathrm{D}(f(U_n), c) = 1 \mid c = b(U_n)\right] \\ + \Pr[c = \overline{b(U_n)}] \cdot \Pr[\mathrm{D}(f(U_n), c) = 0 \mid c = \overline{b(U_n)}]\end{aligned}$$

OWP to PRG cont.

- $\Pr[\mathsf{D}(f(U_n), b(U_n)) = 1] = \delta + \varepsilon'$
- $\Pr[\mathsf{D}(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon'$

Algorithm 9 (P)

Input: $y \in \{0,1\}^n$

1. Flip a random coin $c \leftarrow \{0,1\}$.
2. If $\mathsf{D}(y, c) = 1$ output c, otherwise, output $\overline{c}$.

- It follows that

$$\begin{aligned}\Pr\left[\mathsf{P}(f(U_n)) = b(U_n)\right] &= \Pr[c = b(U_n)] \cdot \Pr\left[\mathsf{D}(f(U_n), c) = 1 \mid c = b(U_n)\right] \\ &+ \Pr[c = \overline{b(U_n)}] \cdot \Pr[\mathsf{D}(f(U_n), c) = 0 \mid c = \overline{b(U_n)}] \\ &= \frac{1}{2} \cdot (\delta + \varepsilon') + \frac{1}{2}(1 - \delta + \varepsilon') = \frac{1}{2} + \varepsilon'.\end{aligned}$$

Part IV

PRG from Regular OWF

Computational notions of entropy

Definition 10

X has (s, ε)-pseudoentropy at least k, if $\exists$ rv Y with $\mathsf{H}(Y) \geq k$ and $\Delta^{\mathsf{D}}(X, Y) \leq \varepsilon$ for any s-size D.
(s, ε)-pseudo min/Reiny -entropy are analogously defined.

Computational notions of entropy

Definition 10

X has (s, ε)-pseudoentropy at least k, if $\exists$ rv Y with $\mathrm{H}(Y) \geq k$ and $\Delta^{\mathsf{D}}(X, Y) \leq \varepsilon$ for any s-size D.
(s, ε)-pseudo min/Reiny -entropy are analogously defined.

- Example

Computational notions of entropy

Definition 10

X has (s, ε)-pseudoentropy at least k, if $\exists$ rv Y with $\mathrm{H}(Y) \geq k$ and $\Delta^{\mathsf{D}}(X, Y) \leq \varepsilon$ for any s-size D.
(s, ε)-pseudo min/Reiny -entropy are analogously defined.

- Example
- Repeated sampling

Computational notions of entropy

Definition 10

X has (s, ε)-pseudoentropy at least k, if $\exists$ rv Y with $\mathrm{H}(Y) \geq k$ and $\Delta^{\mathrm{D}}(X, Y) \leq \varepsilon$ for any s-size D.
(s, ε)-pseudo min/Reiny -entropy are analogously defined.

- Example
- Repeated sampling
- Non-monotonicity

Computational notions of entropy

Definition 10

X has (s, ε)-pseudoentropy at least k, if $\exists$ rv Y with $\mathrm{H}(Y) \geq k$ and $\Delta^{\mathsf{D}}(X, Y) \leq \varepsilon$ for any s-size D.
(s, ε)-pseudo min/Reiny -entropy are analogously defined.

- Example
- Repeated sampling
- Non-monotonicity
- Ensembles

Computational notions of entropy

Definition 10

X has (s,ε)-pseudoentropy at least k, if $\exists$ rv Y with $\mathrm{H}(Y) \geq k$ and $\Delta^{\mathsf{D}}(X,Y) \leq \varepsilon$ for any s-size D.
(s,ε)-pseudo min/Reiny -entropy are analogously defined.

- Example
- Repeated sampling
- Non-monotonicity
- Ensembles
- In the following we will simply write (s,ε)-entropy, etc

High entropy OWF from regular OWF

Claim 11

Let $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ be a 2^k-regular (s,ε)-one-way, let $\mathcal{H} = \{h\colon \{0,1\}^n \mapsto \{0,1\}^{m=k+\lceil \log n \rceil}\}$ be 2-universal family, and let $g(h,x) = (g(x), h, h(x))$. Then

1. $\mathrm{H}_2(g(U_n,H)) \geq 2n - \frac{1}{n}$, for $H \leftarrow \mathcal{H}$.
2. g is $(\Theta(s\varepsilon^2/n), 2\varepsilon)$-one-way.

- k and m and $\mathcal{H}$ are parameterized by of n
- We assume $\log|\mathcal{H}| = n$ and $s \geq n$

g has high Renyi entropy

g has high Renyi entropy

$$
\begin{aligned}
\mathrm{CP}(g(U_n, H)) &:= \Pr_{w,w' \leftarrow \{0,1\}^n \times \mathcal{H}}[g(w) = g(w')] \\
&= \Pr_{h,h' \leftarrow \mathcal{H}}[h = h'] \cdot \Pr_{(x,x') \leftarrow (\{0,1\}^n)^2}[f(x) = f(x')] \\
&\cdot \Pr_{h \leftarrow \mathcal{H}; (x,x') \leftarrow (\{0,1\}^n)^2}[h(x) = h(x') \mid f(x) = f(x')]
\end{aligned}
$$

g has high Renyi entropy

$$
\begin{aligned}
\mathrm{CP}(g(U_n, H)) &:= \Pr_{w,w' \leftarrow \{0,1\}^n \times \mathcal{H}}[g(w) = g(w')] \\
&= \Pr_{h,h' \leftarrow \mathcal{H}}[h = h'] \cdot \Pr_{(x,x') \leftarrow (\{0,1\}^n)^2}[f(x) = f(x')] \\
&\cdot \Pr_{h \leftarrow \mathcal{H}; (x,x') \leftarrow (\{0,1\}^n)^2}[h(x) = h(x') \mid f(x) = f(x')] \\
&= \mathrm{CP}(\mathrm{H}) \cdot \mathrm{CP}(f(U_n)) \cdot (2^{-k} + (1 - 2^{-k}) \cdot 2^{-m})
\end{aligned}
$$

g has high Renyi entropy

$$
\begin{aligned}
\mathrm{CP}(g(U_n, H)) &:= \Pr_{w,w' \leftarrow \{0,1\}^n \times \mathcal{H}}[g(w) = g(w')] \\
&= \Pr_{h,h' \leftarrow \mathcal{H}}[h = h'] \cdot \Pr_{(x,x') \leftarrow (\{0,1\}^n)^2}[f(x) = f(x')] \\
&\cdot \Pr_{h \leftarrow \mathcal{H}; (x,x') \leftarrow (\{0,1\}^n)^2}[h(x) = h(x') \mid f(x) = f(x')] \\
&= \mathrm{CP}(\mathrm{H}) \cdot \mathrm{CP}(f(U_n)) \cdot (2^{-k} + (1 - 2^{-k}) \cdot 2^{-m}) \\
&\leq \mathrm{CP}(H) \cdot \mathrm{CP}(f(U_n)) \cdot (2^{-k} + 2^{-m})
\end{aligned}
$$

g has high Renyi entropy

$$
\begin{aligned}
\mathrm{CP}(g(U_n, H)) &:= \Pr_{w,w' \leftarrow \{0,1\}^n \times \mathcal{H}}[g(w) = g(w')] \\
&= \Pr_{h,h' \leftarrow \mathcal{H}}[h = h'] \cdot \Pr_{(x,x') \leftarrow (\{0,1\}^n)^2}[f(x) = f(x')] \\
&\cdot \Pr_{h \leftarrow \mathcal{H}; (x,x') \leftarrow (\{0,1\}^n)^2}[h(x) = h(x') \mid f(x) = f(x')] \\
&= \mathrm{CP}(\mathrm{H}) \cdot \mathrm{CP}(f(U_n)) \cdot (2^{-k} + (1 - 2^{-k}) \cdot 2^{-m}) \\
&\leq \mathrm{CP}(H) \cdot \mathrm{CP}(f(U_n)) \cdot (2^{-k} + 2^{-m}) \\
&\leq \mathrm{CP}(H) \cdot (2^{-n} + 2^{-n-\log n}) = \mathrm{CP}(H) \cdot \mathrm{CP}(U_n) \cdot (1 + \frac{1}{n}).
\end{aligned}
$$

g has high Renyi entropy

$$
\begin{aligned}
\mathrm{CP}(g(U_n,H)) &:= \Pr_{w,w'\leftarrow\{0,1\}^n\times\mathcal{H}}[g(w)=g(w')] \\
&= \Pr_{h,h'\leftarrow\mathcal{H}}[h=h'] \cdot \Pr_{(x,x')\leftarrow(\{0,1\}^n)^2}[f(x)=f(x')] \\
&\cdot \Pr_{h\leftarrow\mathcal{H};(x,x')\leftarrow(\{0,1\}^n)^2}[h(x)=h(x') \mid f(x)=f(x')] \\
&= \mathrm{CP}(\mathrm{H}) \cdot \mathrm{CP}(f(U_n)) \cdot (2^{-k} + (1-2^{-k}) \cdot 2^{-m}) \\
&\le \mathrm{CP}(H) \cdot \mathrm{CP}(f(U_n)) \cdot (2^{-k} + 2^{-m}) \\
&\le \mathrm{CP}(H) \cdot (2^{-n} + 2^{-n-\log n}) = \mathrm{CP}(H) \cdot \mathrm{CP}(U_n) \cdot (1+\frac{1}{n}).
\end{aligned}
$$

Hence, $\mathrm{H}_2(g(U_n,H)) \ge \mathrm{H}_2(\mathcal{H}) + \mathrm{H}_2(U_n) + \log\frac{1}{1+\frac{1}{n}} \ge n + n - \frac{1}{n}$.

g has high Renyi entropy

$$\begin{aligned}
\mathrm{CP}(g(U_n, H)) &:= \Pr_{w,w' \leftarrow \{0,1\}^n \times \mathcal{H}}[g(w) = g(w')] \\
&= \Pr_{h,h' \leftarrow \mathcal{H}}[h = h'] \cdot \Pr_{(x,x') \leftarrow (\{0,1\}^n)^2}[f(x) = f(x')] \\
&\cdot \Pr_{h \leftarrow \mathcal{H}; (x,x') \leftarrow (\{0,1\}^n)^2}[h(x) = h(x') \mid f(x) = f(x')] \\
&= \mathrm{CP}(\mathrm{H}) \cdot \mathrm{CP}(f(U_n)) \cdot (2^{-k} + (1 - 2^{-k}) \cdot 2^{-m}) \\
&\leq \mathrm{CP}(H) \cdot \mathrm{CP}(f(U_n)) \cdot (2^{-k} + 2^{-m}) \\
&\leq \mathrm{CP}(H) \cdot (2^{-n} + 2^{-n-\log n}) = \mathrm{CP}(H) \cdot \mathrm{CP}(U_n) \cdot (1 + \frac{1}{n}).
\end{aligned}$$

Hence, $\mathrm{H}_2(g(U_n, H)) \geq \mathrm{H}_2(\mathcal{H}) + \mathrm{H}_2(U_n) + \log \frac{1}{1+\frac{1}{n}} \geq n + n - \frac{1}{n}$.

g is one-way

g is one-way

Let A be an s'-size algorithm that inverts g w.p ε' and let $\ell = k - \left\lceil 2 \log \frac{1}{\varepsilon'} \right\rceil$.

g is one-way

Let A be an s'-size algorithm that inverts g w.p ε' and let $\ell = k - \left\lceil 2\log\frac{1}{\varepsilon'}\right\rceil$.

Consider the following inverter for f

Algorithm 12 (B)

Input: $y \in \{0,1\}^n$.
Return $\mathsf{D}(y,h,z)$, for $h \leftarrow \mathcal{H}$ and $z \leftarrow \{0,1\}^\ell$.

Algorithm 13 (D)

Input: $y \in \{0,1\}^n$, $h \in \mathcal{H}$ and $z_1 \in \{0,1\}^\ell$.
For all $z_2 \in \{0,1\}^{m-\ell}$:

1. Let $(x,h) = \mathsf{A}(y,h,z_1 \circ z_2)$.
2. If $f(x) = y$, return x.

g is one-way

Let A be an s'-size algorithm that inverts g w.p ε' and let $\ell = k - \left\lceil 2\log\frac{1}{\varepsilon'}\right\rceil$.

Consider the following inverter for f

Algorithm 12 (B)

Input: $y \in \{0,1\}^n$.
Return $\mathsf{D}(y,h,z)$, for $h \leftarrow \mathcal{H}$ and $z \leftarrow \{0,1\}^\ell$.

Algorithm 13 (D)

Input: $y \in \{0,1\}^n$, $h \in \mathcal{H}$ and $z_1 \in \{0,1\}^\ell$.

For all $z_2 \in \{0,1\}^{m-\ell}$:

1. Let $(x,h) = \mathsf{A}(y,h,z_1 \circ z_2)$.
2. If $f(x) = y$, return x.

- B's size is $((s' + O(n)) \cdot 2^{2\log\frac{1}{\varepsilon'} + \log n + 1} = \Theta(s'n/\varepsilon^2)$

g is one-way

Let A be an s'-size algorithm that inverts g w.p ε' and let $\ell = k - \left\lceil 2\log\frac{1}{\varepsilon'} \right\rceil$.

Consider the following inverter for f

Algorithm 12 (B)

Input: $y \in \{0,1\}^n$.
Return $\mathsf{D}(y, h, z)$, for $h \leftarrow \mathcal{H}$ and $z \leftarrow \{0,1\}^\ell$.

Algorithm 13 (D)

Input: $y \in \{0,1\}^n$, $h \in \mathcal{H}$ and $z_1 \in \{0,1\}^\ell$.
For all $z_2 \in \{0,1\}^{m-\ell}$:

1. Let $(x, h) = \mathsf{A}(y, h, z_1 \circ z_2)$.
2. If $f(x) = y$, return x.

- B's size is $((s' + O(n)) \cdot 2^{2\log\frac{1}{\varepsilon'} + \log n + 1} = \Theta(s'n/\varepsilon^2)$
- $\Pr_{x \leftarrow \{0,1\}^n; h \leftarrow \mathcal{H}}\left[\mathsf{D}(f(x), h, h(x)_{1,\ldots,\ell}) \in f^{-1}(f(x))\right] \geq \varepsilon'$

g is one-way, cont.

We saw that

$$\Pr_{x \leftarrow \{0,1\}^n; h \leftarrow \mathcal{H}} \left[\mathsf{D}(f(x), h, h(x)_{1,\ldots,\ell}) \in f^{-1}(f(x)) \right] \geq \varepsilon' \tag{1}$$

g is one-way, cont.

We saw that

$$\Pr_{x \leftarrow \{0,1\}^n; h \leftarrow \mathcal{H}} \left[\mathsf{D}(f(x), h, h(x)_{1,\ldots,\ell}) \in f^{-1}(f(x)) \right] \geq \varepsilon' \qquad (1)$$

By the leftover hash lemma

$$\mathsf{SD}((f(x), h, h(x)_{1,\ldots,\ell})_{x \leftarrow \{0,1\}, h \leftarrow \mathcal{H}}, (f(x), h, U_\ell)_{x \leftarrow \{0,1\}, h \leftarrow \mathcal{H}}) \leq \varepsilon'/2 \qquad (2)$$

g is one-way, cont.

We saw that

$$\Pr_{x \leftarrow \{0,1\}^n; h \leftarrow \mathcal{H}} \left[\mathsf{D}(f(x), h, h(x)_{1,\ldots,\ell}) \in f^{-1}(f(x)) \right] \geq \varepsilon' \tag{1}$$

By the leftover hash lemma

$$\mathsf{SD}((f(x), h, h(x)_{1,\ldots,\ell})_{x \leftarrow \{0,1\}, h \leftarrow \mathcal{H}}, (f(x), h, U_\ell)_{x \leftarrow \{0,1\}, h \leftarrow \mathcal{H}}) \leq \varepsilon'/2 \tag{2}$$

Hence,

$$\Pr_{x \leftarrow \{0,1\}^n} \left[\mathsf{B}(f(x)) \in f^{-1}(f(x)) \right] \geq \varepsilon' - \varepsilon'/2 = \varepsilon'/2.$$

The generator

Claim 14

Let $g\colon \{0,1\}^n \mapsto \{0,1\}^m$ be a function with $\mathrm{H}_2(f(U_n)) \geq n - \frac{1}{2}$, and let b be (s,ε)-hardcore predicate for g. Then $v(U_n) = (g(U_n), b(U_n))$ has (s,ε)-Renyi-entropy $n + \frac{1}{2}$.

The generator

Claim 14

Let $g\colon \{0,1\}^n \mapsto \{0,1\}^m$ be a function with $\mathrm{H}_2(f(U_n)) \geq n - \frac{1}{2}$, and let b be (s,ε)-hardcore predicate for g. Then $v(U_n) = (g(U_n), b(U_n))$ has (s,ε)-Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

The generator

Claim 14

Let $g\colon \{0,1\}^n \mapsto \{0,1\}^m$ be a function with $\mathsf{H}_2(f(U_n)) \geq n - \frac{1}{2}$, and let b be (s,ε)-hardcore predicate for g. Then $v(U_n) = (g(U_n), b(U_n))$ has (s,ε)-Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

We call such v a pseudo Renyi-entropy generator.

The generator

Claim 14

Let $g \colon \{0,1\}^n \mapsto \{0,1\}^m$ be a function with $\mathrm{H}_2(f(U_n)) \geq n - \frac{1}{2}$, and let b be (s, ε)-hardcore predicate for g. Then $v(U_n) = (g(U_n), b(U_n))$ has (s, ε)-Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

We call such v a pseudo Renyi-entropy generator.

Claim 15

The function $v^n(x_1, \ldots, x_n) = (v(x_1), \ldots, v(x_n))$ has $(s - n^2, n\varepsilon)$-Renyi-entropy $n^2 + \frac{n}{2}$.

The generator

Claim 14

Let $g\colon \{0,1\}^n \mapsto \{0,1\}^m$ be a function with $\mathrm{H}_2(f(U_n)) \geq n - \frac{1}{2}$, and let b be (s,ε)-hardcore predicate for g. Then $v(U_n) = (g(U_n), b(U_n))$ has (s,ε)-Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

We call such v a pseudo Renyi-entropy generator.

Claim 15

The function $v^n(x_1,\ldots,x_n) = (v(x_1),\ldots,v(x_n))$ has $(s - n^2, n\varepsilon)$-Renyi-entropy $n^2 + \frac{n}{2}$.

Proof:

The generator

Claim 14

Let $g\colon \{0,1\}^n \mapsto \{0,1\}^m$ be a function with $\mathrm{H}_2(f(U_n)) \geq n - \frac{1}{2}$, and let b be (s,ε)-hardcore predicate for g. Then $v(U_n) = (g(U_n), b(U_n))$ has (s,ε)-Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

We call such v a pseudo Renyi-entropy generator.

Claim 15

The function $v^n(x_1,\ldots,x_n) = (v(x_1),\ldots,v(x_n))$ has $(s - n^2, n\varepsilon)$-Renyi-entropy $n^2 + \frac{n}{2}$.

Proof:

- Let Z be a rv with $\mathrm{H}_2(Z) \geq n + \frac{1}{2}$ such that Z and $v(U_n)$ are (s,ε) indistinguishable.

The generator

Claim 14

Let $g\colon \{0,1\}^n \mapsto \{0,1\}^m$ be a function with $\mathrm{H}_2(f(U_n)) \geq n - \frac{1}{2}$, and let b be (s,ε)-hardcore predicate for g. Then $v(U_n) = (g(U_n), b(U_n))$ has (s,ε)-Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

We call such v a pseudo Renyi-entropy generator.

Claim 15

The function $v^n(x_1,\ldots,x_n) = (v(x_1),\ldots,v(x_n))$ has $(s-n^2, n\varepsilon)$-Renyi-entropy $n^2 + \frac{n}{2}$.

Proof:

- Let Z be a rv with $\mathrm{H}_2(Z) \geq n + \frac{1}{2}$ such that Z and $v(U_n)$ are (s,ε) indistinguishable.
- $\mathrm{H}_2(Z^n) \geq n^2 + \frac{n}{2}$

The generator

Claim 14

Let $g \colon \{0,1\}^n \mapsto \{0,1\}^m$ be a function with $\mathrm{H}_2(f(U_n)) \geq n - \frac{1}{2}$, and let b be (s,ε)-hardcore predicate for g. Then $v(U_n) = (g(U_n), b(U_n))$ has (s,ε)-Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

We call such v a pseudo Renyi-entropy generator.

Claim 15

The function $v^n(x_1, \ldots, x_n) = (v(x_1), \ldots, v(x_n))$ has $(s - n^2, n\varepsilon)$-Renyi-entropy $n^2 + \frac{n}{2}$.

Proof:

- Let Z be a rv with $\mathrm{H}_2(Z) \geq n + \frac{1}{2}$ such that Z and $v(U_n)$ are (s,ε) indistinguishable.
- $\mathrm{H}_2(Z^n) \geq n^2 + \frac{n}{2}$
- Z^n and $v^n(U_n^n)$ are $(s - n^2, n\varepsilon)$ indistinguishable

The generator

Claim 14

Let $g\colon \{0,1\}^n \mapsto \{0,1\}^m$ be a function with $\mathsf{H}_2(f(U_n)) \geq n - \frac{1}{2}$, and let b be (s,ε)-hardcore predicate for g. Then $v(U_n) = (g(U_n), b(U_n))$ has (s,ε)-Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

We call such v a pseudo Renyi-entropy generator.

Claim 15

The function $v^n(x_1,\ldots,x_n) = (v(x_1),\ldots,v(x_n))$ has $(s - n^2, n\varepsilon)$-Renyi-entropy $n^2 + \frac{n}{2}$.

Proof:

- Let Z be a rv with $\mathsf{H}_2(Z) \geq n + \frac{1}{2}$ such that Z and $v(U_n)$ are (s,ε) indistinguishable.
- $\mathsf{H}_2(Z^n) \geq n^2 + \frac{n}{2}$
- Z^n and $v^n(U_n^n)$ are $(s - n^2, n\varepsilon)$ indistinguishable

The generator

Claim 14

Let $g\colon \{0,1\}^n \mapsto \{0,1\}^m$ be a function with $\mathrm{H}_2(f(U_n)) \geq n - \frac{1}{2}$, and let b be (s,ε)-hardcore predicate for g. Then $v(U_n) = (g(U_n), b(U_n))$ has (s,ε)-Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

We call such v a pseudo Renyi-entropy generator.

Claim 15

The function $v^n(x_1,\ldots,x_n) = (v(x_1),\ldots,v(x_n))$ has $(s - n^2, n\varepsilon)$-Renyi-entropy $n^2 + \frac{n}{2}$.

Proof:

- Let Z be a rv with $\mathrm{H}_2(Z) \geq n + \frac{1}{2}$ such that Z and $v(U_n)$ are (s,ε) indistinguishable.
- $\mathrm{H}_2(Z^n) \geq n^2 + \frac{n}{2}$
- Z^n and $v^n(U_n^n)$ are $(s - n^2, n\varepsilon)$ indistinguishable

The generator cont.

The generator cont.

Claim 16

Let $\mathcal{H}\colon \{0,1\}^{n^2+n} \mapsto \{0,1\}^{n^2+n/4}$ be an 2-universal family and let $G\colon \{0,1\}^n \times \mathcal{H}$ defined by $G(x_1,\ldots,x_n,h) = (h, h(v^n(x_1,\ldots,x_n)))$. Then $G(H,U_n^n)$ is $(s - ns_v - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for $\mathcal{H}$.

The generator cont.

Claim 16

Let $\mathcal{H}\colon \{0,1\}^{n^2+n} \mapsto \{0,1\}^{n^2+n/4}$ be an 2-universal family and let $G\colon \{0,1\}^n \times \mathcal{H}$ defined by $G(x_1,\ldots,x_n,h) = (h, h(v^n(x_1,\ldots,x_n)))$. Then $G(H,U_n^n)$ is $(s - ns_v - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for $\mathcal{H}$.

Corollary 17

If f and b and $\mathcal{H}$ (?) are poly-time computable, then G is a $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$-PRG.

The generator cont.

Claim 16

Let $\mathcal{H}\colon \{0,1\}^{n^2+n} \mapsto \{0,1\}^{n^2+n/4}$ be an 2-universal family and let $G\colon \{0,1\}^n \times \mathcal{H}$ defined by $G(x_1,\ldots,x_n,h) = (h, h(v^n(x_1,\ldots,x_n)))$. Then $G(H, U_n^n)$ is $(s - ns_v - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for $\mathcal{H}$.

Corollary 17

If f and b and $\mathcal{H}$ (?) are poly-time computable, then G is a $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$-PRG.

Proof: (of claim)

The generator cont.

Claim 16

Let $\mathcal{H}\colon \{0,1\}^{n^2+n} \mapsto \{0,1\}^{n^2+n/4}$ be an 2-universal family and let $G\colon \{0,1\}^n \times \mathcal{H}$ defined by $G(x_1,\ldots,x_n,h) = (h, h(v^n(x_1,\ldots,x_n)))$. Then $G(H,U_n^n)$ is $(s - ns_v - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for $\mathcal{H}$.

Corollary 17

If f and b and $\mathcal{H}$ (?) are poly-time computable, then G is a $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$-PRG.

Proof: (of claim)

- By the leftover hash lemma $\mathrm{SD}((H, H(Z^n)), (H, U_{n^2+n/4})) \leq 2^{-n/4}$

The generator cont.

Claim 16

Let $\mathcal{H}\colon \{0,1\}^{n^2+n} \mapsto \{0,1\}^{n^2+n/4}$ be an 2-universal family and let $G\colon \{0,1\}^n \times \mathcal{H}$ defined by $G(x_1,\ldots,x_n,h) = (h, h(v^n(x_1,\ldots,x_n)))$. Then $G(H,U_n^n)$ is $(s - ns_v - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for $\mathcal{H}$.

Corollary 17

If f and b and $\mathcal{H}$ (?) are poly-time computable, then G is a $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$-PRG.

Proof: (of claim)

- By the leftover hash lemma $\mathrm{SD}((H, H(Z^n)), (H, U_{n^2+n/4})) \le 2^{-n/4}$
- Let D be an s'-size algorithm that distinguishes $G(U_n^n, H)$ from $(H, U_{n^2+n/4})$ with advantage $\varepsilon' + 2^{-n/4}$

The generator cont.

Claim 16

Let $\mathcal{H}\colon \{0,1\}^{n^2+n} \mapsto \{0,1\}^{n^2+n/4}$ be an 2-universal family and let $G\colon \{0,1\}^n \times \mathcal{H}$ defined by $G(x_1,\ldots,x_n,h) = (h, h(v^n(x_1,\ldots,x_n)))$. Then $G(H, U_n^n)$ is $(s - ns_v - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for $\mathcal{H}$.

Corollary 17

If f and b and $\mathcal{H}$ (?) are poly-time computable, then G is a $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$-PRG.

Proof: (of claim)

- By the leftover hash lemma $\mathrm{SD}((H, H(Z^n)), (H, U_{n^2+n/4})) \leq 2^{-n/4}$
- Let D be an s'-size algorithm that distinguishes $G(U_n^n, H)$ from $(H, U_{n^2+n/4})$ with advantage $\varepsilon' + 2^{-n/4}$
- Hence, $\exists$ $(s' + s_{\mathcal{H}})$-size algorithm that distinguishes $G(U_n^n, H)$ from $(H, H(Z^n))$ with advantage ε'

The generator cont.

Claim 16

Let $\mathcal{H}\colon \{0,1\}^{n^2+n} \mapsto \{0,1\}^{n^2+n/4}$ be an 2-universal family and let $G\colon \{0,1\}^n \times \mathcal{H}$ defined by $G(x_1,\ldots,x_n,h) = (h, h(v^n(x_1,\ldots,x_n)))$. Then $G(H, U_n^n)$ is $(s - ns_v - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for $\mathcal{H}$.

Corollary 17

If f and b and $\mathcal{H}$ (?) are poly-time computable, then G is a $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$-PRG.

Proof: (of claim)

- By the leftover hash lemma $\mathrm{SD}((H, H(Z^n)), (H, U_{n^2+n/4})) \leq 2^{-n/4}$
- Let D be an s'-size algorithm that distinguishes $G(U_n^n, H)$ from $(H, U_{n^2+n/4})$ with advantage $\varepsilon' + 2^{-n/4}$
- Hence, $\exists$ $(s' + s_{\mathcal{H}})$-size algorithm that distinguishes $G(U_n^n, H)$ from $(H, H(Z^n))$ with advantage ε'
- Hence $s' \leq s - n^2 - s_{\mathcal{H}} \implies \varepsilon' \leq n\varepsilon$.

The generator cont.

Claim 16

Let $\mathcal{H}\colon \{0,1\}^{n^2+n} \mapsto \{0,1\}^{n^2+n/4}$ be an 2-universal family and let $G\colon \{0,1\}^n \times \mathcal{H}$ defined by $G(x_1,\ldots,x_n,h) = (h, h(v^n(x_1,\ldots,x_n)))$. Then $G(H,U_n^n)$ is $(s - ns_v - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for $\mathcal{H}$.

Corollary 17

If f and b and $\mathcal{H}$ (?) are poly-time computable, then G is a $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$-PRG.

Proof: (of claim)

- By the leftover hash lemma $\mathsf{SD}((H, H(Z^n)), (H, U_{n^2+n/4})) \le 2^{-n/4}$
- Let D be an s'-size algorithm that distinguishes $G(U_n^n, H)$ from $(H, U_{n^2+n/4})$ with advantage $\varepsilon' + 2^{-n/4}$
- Hence, $\exists$ $(s' + s_{\mathcal{H}})$-size algorithm that distinguishes $G(U_n^n, H)$ from $(H, H(Z^n))$ with advantage ε'
- Hence $s' \le s - n^2 - s_{\mathcal{H}} \implies \varepsilon' \le n\varepsilon$.

The generator cont.

Claim 16

Let $\mathcal{H}\colon \{0,1\}^{n^2+n} \mapsto \{0,1\}^{n^2+n/4}$ be an 2-universal family and let $G\colon \{0,1\}^n \times \mathcal{H}$ defined by $G(x_1,\ldots,x_n,h) = (h, h(v^n(x_1,\ldots,x_n)))$. Then $G(H, U_n^n)$ is $(s - ns_v - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for $\mathcal{H}$.

Corollary 17

If f and b and $\mathcal{H}$ (?) are poly-time computable, then G is a $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$-PRG.

Proof: (of claim)

- By the leftover hash lemma $\mathrm{SD}((H, H(Z^n)), (H, U_{n^2+n/4})) \le 2^{-n/4}$
- Let D be an s'-size algorithm that distinguishes $G(U_n^n, H)$ from $(H, U_{n^2+n/4})$ with advantage $\varepsilon' + 2^{-n/4}$
- Hence, $\exists$ $(s' + s_{\mathcal{H}})$-size algorithm that distinguishes $G(U_n^n, H)$ from $(H, H(Z^n))$ with advantage ε'
- Hence $s' \le s - n^2 - s_{\mathcal{H}} \implies \varepsilon' \le n\varepsilon$.

Remarks

- PRG “length extension"

Remarks

- PRG "length extension"
- PRG from any OWF